코딩 없는 임베디드 에뮬레이터 생성 도구 설계 및 프로토타이핑

김소임*, 김영제*, 나영국*
*서울시립대학교 전자전기컴퓨터학과
e-mail : soimkim12@gmail.com, chozekun@naver.com, ygra123@gmail.com

Prototyping and Designing of an embedded emulator creating tool without coding

So-im Kim*, Young je Kim*, Young-gook Ra*
*Dept. of Electronic& Electrical Computer Engineering, University of Seoul

요 약

본 논문에서는 임베디드 소프트웨어 개발 시 .Net Micro Framework를 이용하여 디바이스들의 통신 인터페이스를 소프트웨어 개발자가 좀 더 간단하게 코딩 가능하게 하며 소프트웨어 개발자가 이를 디자이너를 이용하여 그래픽하게 에뮬레이터를 디자인할 수 있도록 임베디드 에뮬레이터 생성 방법을 제시하고 디자인 된 에뮬레이터를 기반으로 자동으로 코딩 없이 에뮬레이터를 생성하는 프로그램을 프로토타이핑 한다. 이때, 소프트웨어 개발자가 그래픽하게 생성한 에뮬레이터 디자인으로 하드웨어 개발자와의 의사소통이 보다 원활해질 수 있다. 또한 본 논문에서 제시하는 임베디드 에뮬레이터 생성 프로그램을 이용하면 추가적인 코딩 작업 없이 디자인 된 에뮬레이터를 기반으로 에뮬레이터가 자동 생성된다. 이로 인하여 .NET Micro Framework에서 제공하는 확장 가능 에뮬레이터와 달리 에뮬레이터를 생성하는데 시간과 노력을 절약할 수 있다. 또한 기존의 개발 과정에서는 기계가 모두 준비되어야 테스팅이 가능했던 것과 달리 기계 없이 에뮬레이터로 테스팅이 가능해짐으로써 개발 기간을 단축시킬 수 있다. 임베디드 에뮬레이터를 이용하면 하드웨어와의 통합 테스트에서 버그의 원인이 하드웨어적인 결함인지 소프트웨어적인 결함인지를 판단 가능하게 하여 디버깅이 용이하게 되며 실제 동작과 원하는 동작을 비교해 볼 수 있다.

1. 서론

임베디드 시스템 개발 시 가장 큰 딜레마는 개발된 사례가 있는지 또는 BSP(Board Support Packages)가 제공되는지 여부이다. 이처럼 BSP의 여부가 중요한 이유는 임베디드 소프트웨어 개발에 있어서 디바이스 드라이버가 최고의 난관이기 때문이다. 디바이스 드라이버는 운영체제 내에서 응용프로그램과 디바이스를 연결시켜주는 프로그램이며 임베디드 시스템 구축 시 반드시 필요하다. 하지만 디바이스 드라이버는 하드웨어 정보가 적절하게 입력되어야 하기 때문에 개발이 어렵고 높은 품질을 유지하기 또한 어렵다. 이러한 디바이스 드라이버로 .NET Micro Framework를 이용하면 특정 하드웨어와 관련된 불필요하게 신경 써야 할 일들로부터 소프트웨어 개발자를 분리해주게 된다. 또한, 에뮬레이터가 제공되는 안드로이드 또는 아 이폰과 같은 임베디드 소프트웨어 개발은 소프트웨어 개발자가 하드웨어에 관련된 많은 지식을 필요로 하지 않으며 실제 기계 없이도 편리하게 테스트 해볼 수 있다는 장점이 있다.

임베디드 에뮬레이터로 .NET Micro Framework에서 확장 가능 에뮬레이터를 제공하지만 이는 간단한 컴포넌트의 추가에도 코딩을 필요로 하여 에뮬레이터 개발에 많은 시간과 노력을 낭비하게 한다. 이에 본 논문에서는 .NET Micro Framework를 사용하여 임베디드 소프트웨어 개발에 있어서 소프트웨어 개발자가 그래픽하게 에뮬레이터를 디자인하고 이를 기반으로 코딩 없이 자동으로 에뮬레이터를 생성하는 프로그램을 프로토타이핑 하였다.

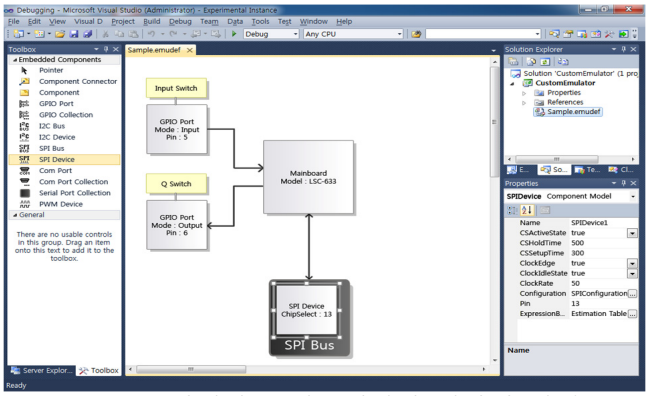
2. 시스템 구현

2.1 임베디드 에뮬레이터 디자이너

코딩 없이 임베디드 에뮬레이터를 생성하기 위하여 에뮬레이터 생성 도구를 Visual studio extension을 이용하여 설계하였다. 임베디드 에뮬레이터 프로젝트에서 제공할 툴 박스의 컴포넌트의 목록은 아래와 같다.

Embedded Components
- •Pointer
- •Component Connector
- •Component
- •GPIO Port
- •GPIO Collection
- •I2C Bus
- •I2C Device

•SPI Bus
•SPI Device
•Com Port
•Com Port Collection
•PWM Device

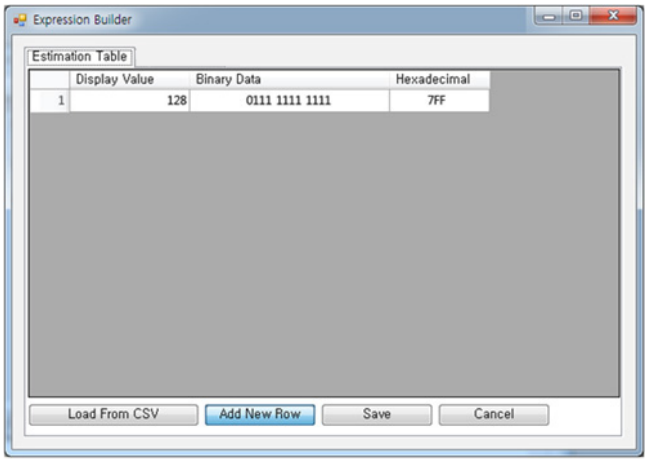


(그림1) 임베디드 에뮬레이터 디자인 화면

임베디드 에뮬레이터를 생성하기 위한 과정은 아래와 같다.

①에뮬레이터 프로젝트 생성.
②디자이너를 이용한 emulator component 정의.
③emulator에서 동작할 application software작성.

SPI와 I2C Device의 경우 Expression Builder를 설정하여 사용자가 에뮬레이터에서 입력하는 값과 실제로 레지스터에 쓰이는 값을 변환시킬 수 있다. 임베디드 에뮬레이터 디자인 화면의 에뮬레이터 컴포넌트(SPI 또는 I2C)를 선택 후 속성 창의 Expression Builder를 클릭하면 Estimation Table이 (그림2)와 같이 생성된다. Estimation table은 Display value, Binary Data, Hexadecimal 세 개의 필드로 이루어져 있다. 각각의 데이터를 입력 후 Add New Row을 클릭하면 새로운 value를 추가 할 수 있고 작업 완료 후에는 Save 버튼을 눌러 저장한다. 에뮬레이터 실행 시 Display Value 값을 해당 포트에 입력하면 Binary Data로 변환 해주게 된다.



(그림 2) Expression Builder

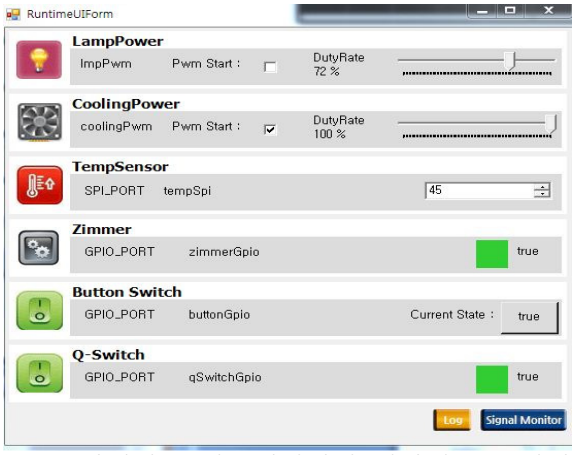
2.2 임베디드 에뮬레이터

임베디드 에뮬레이터라는 이름에도 불구하고 CPU를 실제 에뮬레이션하지 않기 때문에 기술적으로는 시뮬레이터라고 할 수 있다. .NET Micro Framework 에서 제공하는 확장 가능 에뮬레이터와 마찬가지로 최종 하드웨어의 시뮬레이션 도구라고 하는 것이 적절하다. 실제 하드웨어는 장치 드라이버를 사용하지만 임베디드 에뮬레이터는 제공된 윈도우 드라이버를 사용하게 된다.

임베디드 에뮬레이터는 디자이너를 통해 어플리케이션 개발자가 생성한 임베디드 에뮬레이터 디자인 xml파일을 파싱(parsing)하여 코딩 없이 자동으로 임베디드 에뮬레이터를 생성한다. 임베디드 에뮬레이터의 화면은 크게 3가지로 나누어진다.

➢ **Device Monitor**

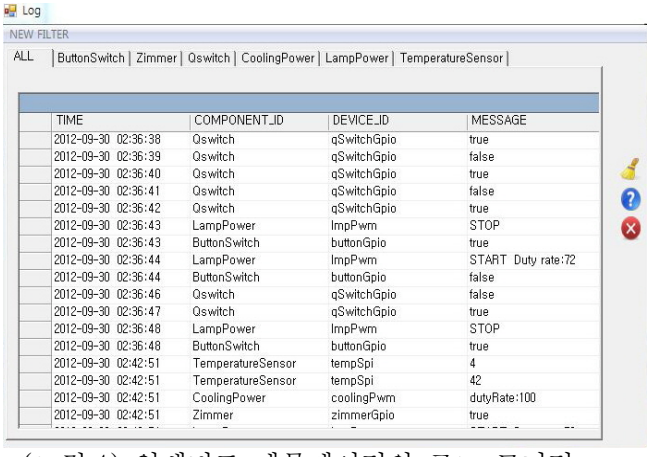
장치들을 정렬하고 장치가 포함하는 포트들을 보여준다. 포트들의 레지스터 값을 확인 할 수 있고 입력할 수 있다.



(그림 3) 임베디드 에뮬레이터의 디바이스 모니터

➢ **Log Monitor**

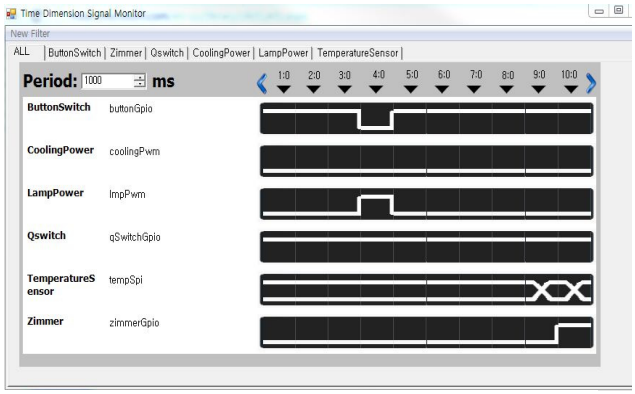
포트들의 이벤트가 발생함에 따라 로그를 확인할 수 있다. add filter 기능을 통하여 로그를 확인하기 원하는 포트만 골라 탭을 생성할 수 있다.



(그림 4) 임베디드 에뮬레이터의 로그 모니터

➢ Signal Monitor

실시간으로 포트들의 레지스터 값이 바뀌는 것을 확인 할 수 있다.
stop, play 기능을 통해 일시 정지가 가능하며 시간대별 신호(signal)을 시간 네비게이터를 통해 확인할 수 있다. 또한 add filter 기능을 통하여 신호를 확인하기 원하는 포트만 골라 화면을 구성할 수 있다.



(그림5) 임베디드 에뮬레이터의 신호 모니터

3. 결론

본 논문에서는 코딩 없이 임베디드 에뮬레이터를 생성하여 임베디드 소프트웨어 개발자로 하여금 하드웨어 설계가 완료 되지 않더라도 가상으로 통합 테스트를 할 수 있도록 하였다. 이로써 소프트웨어에 대한 신뢰성을 보장할 수 있다. 또한 .NET Micro Framework를 사용하여 응용프로그램 개발자가 하드웨어에 관련한 많은 지식 없이도 디바이스 드라이버 부분을 작성 가능하도록 하였다. 에뮬레이터 생성 도구를 이용하여 코딩 없이 그래픽 하게 에뮬레이터를 디자인하기 때문에 에뮬레이터를 생성하기 간단해졌으며 에뮬레이터 디자인으로 소프트웨어 개발자와 하드웨어 개발자 간의 의사소통을 원활하게 도와주었다.

앞으로 추가되어야 할 작업으로 .NET Micro Framework에서 지원 하지 않는 포트에 대한 API가 추가적으로 제공되어야 한다. 또한 Expression Builder에 대하여 Estimation Table 뿐만 아니라 간단한 함수를 작성하고 연결하여 읽고 쓰는 데이터를 변환시킬 수 있도록 하는 작업이 필요하다.

참고문헌

[1] Donald Thompson, Rob S. Miles, “Embedded Programming with the Microsoft .Net Micro Framework”, 2007
[2] Kuhner, Jens, “Expert .NET Micro Framework”, 2008
[3] Randolph, Nick Gardner, David Anderson, Chris, “Professional Visual Studio 2010”, 2010
[4] Tammy Noergaard, “임베디드 시스템 아키텍처”, 2007
[5] 김행곤 외, “임베디드 소프트웨어 공학”, 2006
[6] 박창순 외, “생활 속의 임베디드 소프트웨어”, 2007
[7] 로버트 C 마틴, “UML 실전에서는 이것만 쓴다”, 2010
[8] 최재규, “C# Programming Bible with .Net framework 3.0”, 2007
[9] 박경훈, “HOONS 닷넷과 함께하는 .NET framework 3.0”, 2007
[10] John Schenken, “Professional .NET Framework”, 2001